Hy-dro-gen

Hyphenation patterns from `github:typst/hypher`

```
#import "@preview/hy-dro-gen:0.1.5" as hy
```

`#hy.exists("en")`	`true`
`#hy.exists("xz")`	`false`
`#hy.syllables("hydrogen")`	`("hy", "dro", "gen")`
`#hy.syllables("hydrogène")`	`("hy", "drogène")`
`#hy.syllables("hydrogène", lang: "fr")`	`("hy", "dro", "gène")`
`#hy.syllables("hydrogène", lang: "xz")`	**panic: Invalid language**
`#hy.syllables("hydrogène", lang: "xz", fallback: auto)`	`("hydrogène",)`
`#hy.syllables("hydrogène", lang: "xz", fallback: "fr")`	`("hy", "dro", "gène")`

The following languages are supported:

Code	**Language**		**Code**	**Language**
"af"	Afrikaans		"sq"	Albanian
"be"	Belarusian		"bg"	Bulgarian
"ca"	Catalan		"hr"	Croatian
"cs"	Czech		"da"	Danish
"nl"	Dutch		"en"	English
"et"	Estonian		"fi"	Finnish
"fr"	French		"ka"	Georgian
"de"	German		"el"	Greek
"hu"	Hungarian		"is"	Icelandic
"it"	Italian		"ku"	Kurmanji
"la"	Latin		"lt"	Lithuanian
"mn"	Mongolian		"no"	Norwegian
"nb"	Norwegian		"nn"	Norwegian
"pl"	Polish		"pt"	Portuguese
"ru"	Russian		"sr"	Serbian
"sk"	Slovak		"sl"	Slovenian
"es"	Spanish		"sv"	Swedish
"tr"	Turkish		"tk"	Turkmen
"uk"	Ukrainian			

The above mapping is available via the static dictionary `#hy.languages`.

- exists()
- syllables()

Variables

- languages

exists

Check if a code corresponds to a language that has registered patterns.

See the list of officially supported languages at `github:typst/hypher`

If this function returns `true`, then an invocation of `hyphen-points` with this language is guaranteed to not raise a "Language does not exist" failure.

Parameters

```
exists(lang: str) -> bool
```

lang `str`

2-letter language code.

syllables

Splits a word into syllables according to hyphenation patterns.

Parameters

```
syllables(
  word: string,
  lang: str,
  fallback: str auto none
) -> (..string,)
```

word `string`

Word to split.

lang `str`

Language 2-letter code.

Default: `"en"`

fallback `str` or `auto` or `none`

Determines the behavior in case `lang` is unsupported

- `none`: panics with "Invalid language"
- `auto`: the word is not split at all
- valid 2-letter code: use that instead

Default: `none`

languages `dictionary`

Dictionary of supported codes and languages, in the format:

```
(en: "English", fr: "French", ...)
```